

# Good Trellises for IC Implementation of Viterbi Decoders for Linear Block Codes

Hari T. Moorthy, *Member, IEEE*, Shu Lin, and Gregory T. Uehara

**Abstract**—This paper investigates trellis structures of linear block codes for the integrated circuit (IC) implementation of Viterbi decoders capable of achieving high decoding speed while satisfying a constraint on the structural complexity of the trellis in terms of the maximum number of states at any particular depth. Only uniform sectionalizations of the code trellis diagram are considered. An upper-bound on the number of parallel and structurally identical (or isomorphic) subtrellises in a proper trellis for a code without exceeding the maximum state complexity of the minimal trellis of the code is first derived. Parallel structures of trellises with various section lengths for binary BCH and Reed–Muller (RM) codes of lengths 32 and 64 are analyzed. Next, the complexity of IC implementation of a Viterbi decoder based on an  $L$ -section trellis diagram for a code is investigated. A structural property of a Viterbi decoder called add-compare-select (ACS)-connectivity which is related to state connectivity is introduced. This parameter affects the complexity of wire-routing (interconnections within the IC). The effect of five parameters namely: 1) effective computational complexity; 2) complexity of the ACS-circuit; 3) traceback complexity; 4) ACS-connectivity; and 5) branch complexity of a trellis diagram on the very large scale integration (VLSI) complexity of a Viterbi decoder is investigated. It is shown that an IC implementation of a Viterbi decoder based on a nonminimal trellis requires less area and is capable of operation at higher speed than one based on the minimal trellis when the commonly used ACS-array architecture is considered.

**Index Terms**—ACS-array architecture, trellis diagram, Viterbi decoder.

## I. INTRODUCTION

ANY linear block code can *theoretically* be decoded by applying the Viterbi algorithm to a trellis for the code. Trellises for block codes were first described in [1]–[3]. After Forney's refinement of the structure of these trellises [4], their potential in the *practical* decoding of block codes has been realized by many others who have published extensively on various aspects of the trellis structure of block codes [5]–[25]. In some of the above papers, one goal was to minimize the maximum number of states in the trellis at any depth by considering all possible permutations of the code [6]. For some codes such as Reed–Muller (RM) codes, this optimum

permutation is known [7]. For most others only bounds are known.

Even when the optimum order of bits is known or a good permutation is known (if the optimum order is unknown), previous work has focussed on minimization of the number of computations required for decoding [12], [22], [24]. If the actual decoding is intended to be performed using a stored program approach that executes the operations needed to decode a received vector sequentially, then this approach will lead to the fastest decoding speed. However, if an integrated circuit (IC) implementation is intended, then an alternative approach is more suitable. Given a constraint on the amount of hardware (determined by the number of states and the complexity of branches) in the decoder, decoding must be done as fast as possible; not necessarily with as few computations as possible. To achieve this end, we propose the use of nonminimal trellises with parallel structure in which the maximum state space dimension is not greater than the maximum state space dimension of the minimal trellis of a code. In this paper, certain properties concerning the state connectivity and branch complexity [9] of this nonminimal trellis are derived which demonstrate that the nonminimal trellis implementation would require *less area in an IC* implementation than the corresponding minimal trellis when the ubiquitous add-compare-select (ACS) array architecture [26]–[28] is used for implementation. We caution that if a different architecture as proposed in [27] or [24] is chosen for implementation, then the trellis structure that is best suited will in general be different from the proposed trellis.

The number of decoding operations required by the standard trellis-based Viterbi decoding algorithm depends on the sectionalization of the trellis used for decoding. Most of the previous works focussed on uniform sectionalization of a trellis, each section consists of the same number of code symbols. However, [22] recently showed that nonuniform sectionalization of a trellis often results in less number of decoding operations than uniform sectionalization. They have devised an efficient algorithm for finding optimal sectionalization of a trellis for minimizing the total number of decoding operations required for maximum-likelihood (ML) trellis decoding. Optimal sectionalization of a trellis to minimize computational complexity is also investigated in [24]. In this paper, we only investigate good trellises with uniform sectionalization for IC implementation of Viterbi decoders. Particularly, we are concerned with those structures, such as parallel structure, regularity and state-connectivity that: 1) affects the complexity of wire-routing (interconnections)

Paper approved by S. B. Wicker, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received March 31, 1995; revised November 15, 1995 and April 1, 1996. This paper was supported by NSF Grants NCR 94-15374 and NCR 91-15400 and NASA Grants NAG 5-931 and NAG 5-2938. This paper was presented in part at the IEEE International Symposium on Information Theory, September 17–22, 1995, Whistler Conference Center, Canada.

The authors are with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA.

Publisher Item Identifier S 0090-6778(97)00726-5.

within the IC and chip-size and 2) facilitate parallel and pipeline decoding process to achieve high decoding speed. Since nonuniform sectionalization of a trellis requires less decoding operations, this advantage over uniform sectionalization and other properties definitely should be investigated for IC implementation of Viterbi decoders to achieve high decoding speed. This investigation is beyond the scope of this paper.

Trellises for block codes are often loosely connected. A properly constructed trellis may consist of many parallel and structurally identical (isomorphic) subtrellises of smaller state space dimension without cross-connections between them. Consequently, identical Viterbi decoders of much smaller complexity can be devised to process the subtrellises independently in parallel without internal communication between them. This not only simplifies the IC implementation but also speeds up the decoding process. For example, the (32,16,8) RM code, also an extended BCH code, has a four-section, 64-state minimal trellis diagram, which consists of eight parallel and structurally identical eight-state subtrellises without cross-connections among them. As a result, we can devise eight identical eight-state Viterbi decoders to process the eight subtrellises in parallel without communication between them. At the end, there are eight survivors (one from each subtrellis) and the best one will be chosen as the decoded codeword. This reduces the implementation of a 64-state Viterbi decoder to the implementation of an eight-state decoder and using eight copies of it. This parallel structure reduces the wire-routing and internal communications within IC which reduces chip size and improves decoding speed. If the state and branch complexities of each subtrellis is small and the total number of subtrellises is small, all the subtrellis decoders can be put on a single chip, such as for the (32,16,8) RM code [29]. However, if the state and branch complexities are big, then each subtrellis decoder (or several of them) can be implemented on a single chip. This provides flexibility in chip plan and decoder architecture.

The two fundamental bottlenecks to Viterbi decoding (decoding speed) are the internal communications between ACS units and comparisons of incoming branches (radix-profile) at each state [28], [30]. Properly designed parallel structure in a trellis would overcome these obstacles without exceeding the maximum state space dimension of the minimal trellis. For example, a (64,40,8) RM subcode which is being considered by NASA for high-speed satellite communications has an eight-section 2048-state trellis. This trellis consists of 32 parallel and structurally identical 64-state subtrellises. The last four sections of each subtrellis are a mirror image of the first four sections as shown in Fig. 3. As a result, a bidirectional decoding can be performed. Furthermore, the maximum component of the radix profile for each half subtrellis is only eight. A 64-state subtrellis decoder can be implemented on a single chip in 0.5  $\mu\text{m}$  complementary metal-oxide-semiconductor (CMOS) technology which can operate at a decoding speed of 600 Mps [31]. Other structural properties of the subtrellises for this (64,40,8) RM subcode which simplifies the IC implementation will be discussed later. Parallel structure therefore, offers simplification, flexibility and higher decoding

speed for IC implementation. We must note that the parallel structure does not reduce the total number of single-state processors, i.e., number of ACS's.

In this paper, we investigate trellis structures, particularly the parallel structure, of linear block codes for implementation of Viterbi decoders capable of achieving high decoding speed while satisfying a constraint on the structural complexity of the trellis in terms of the maximum number of states at any depth. Only uniform sectionalizations of the code trellis are considered. The organization of the paper is as follows.

In Section II, using the theory of  $L$ -section minimal trellis diagrams, an upper-bound on the number of parallel isomorphic subtrellises in a proper trellis for a code without exceeding the maximum state space dimension of the minimal trellis of the code is derived. In Section III, we analyze the trellises for all extended BCH and RM codes of lengths 32 and 64. In Section IV, we define parameters related to the complexity of a Viterbi decoder IC using the ACS-array architecture for linear block codes. Section V treats examples and in Section VI we use the results of this paper to design a trellis for a (64,40) RM subcode.

## II. TRELLISES WITH PARALLEL STRUCTURE FOR LINEAR BLOCK CODES WITH CONSTRAINT ON MAXIMUM STATE SPACE DIMENSION

The objective of this section is to show that we can build a trellis for a linear block code  $C$  which is a disjoint union of a certain desired number of parallel isomorphic subtrellises. Although this trellis is not minimal, its state space dimension at every depth is less than or equal to the maximum state space dimension of the minimal trellis. The conditions under which such a trellis construction is possible and an upper-bound on the number of such parallel subtrellises are derived. In some cases, the minimal trellis itself possesses a parallel structure. The number of such parallel subtrellises (if any) in the minimal trellis is derived.

### A. Preliminaries

We consider only binary  $(N, K, d_{\min})$  linear block codes. Let  $L, M$  be positive integers such that  $LM = N$ . The minimal (up to graph isomorphism)  $L$ -section trellis, is a well understood graphical representation of the code [5], [9]. Let the sets of states at the end of each section be denoted  $\{S_0, S_M, S_{2M}, \dots, S_{(L-1)M}, S_{LM}\}$ . We define a sequence  $\{s_0, s_M, \dots, s_{LM}\}$  called the *state complexity profile* (SCP) of the trellis and given by  $s_{iM} = \log_2(|S_{iM}|)$  for  $0 \leq i \leq L$ . The minimal  $L$ -section trellis of a code  $C$  has the property that every component of its SCP is less than or equal to the corresponding component in the SCP of any other proper  $L$ -section trellis for  $C$ . The maximum among the  $N + 1$  components in the SCP of the minimal  $N$ -section trellis ( $L = N, M = 1$ ) for  $C$  is denoted  $s_{\max}(C)$  and we will denote the maximum of the components in the SCP of the minimal  $L$ -section trellis for  $C$  as  $s_{\max,L}(C)$ . For a binary  $N$ -tuple  $\mathbf{v} = (v_1, \dots, v_N)$ , let  $p_{h,h'}[\mathbf{v}]$  denote the  $(h' - h)$ -tuple  $(v_{h+1}, \dots, v_{h'})$  and let  $p_{h,h'}[C] = \{p_{h,h'}[c] : c \in C\}$ . Let  $C_{h,h'}$  be the linear subcode of  $C$  consisting of all

codewords whose components are all zero except for the  $(h' - h)$  components from the  $(h + 1)$ th bit position to the  $h'$ th bit position.

In an  $L$ -section minimal trellis for a block code, there may be a set of parallel branches between two adjacent states. In such a case, we call the entire set of parallel branches a *composite branch*. Each composite branch in the  $i$ th section  $1 \leq i \leq L$ , is made up of  $2^{P_i}$  parallel branches where  $P_i$  is the dimension of the subcode denoted  $C_{(i-1)M, iM}$  [9]. In the  $i$ th section of an  $L$ -section trellis for a linear block code  $1 \leq i \leq L$ , the number of distinct branch metrics that have to be computed is  $2^{D_i}$  where  $D_i$  is the dimension of the subcode  $p_{(i-1)M, iM}(C)$  and this number is much less than the total number of branches.  $D_i$  is the rank of the submatrix formed by  $M$  columns from the  $[(i-1)M + 1]$ th to the  $(iM)$ th column of the generator matrix of the code and is upper-bounded by  $M$ . For  $1 \leq i \leq L$ , let the number of composite branches merging into any state  $s \in S_{iM}$  be  $2^{\delta_{iM}}$  (it is the same for any state in  $S_{iM}$ ). For an  $L$ -section trellis for  $C$ , we define the converging branch profile (CBP) as the ordered sequence  $\{\delta_M, \delta_{2M}, \dots, \delta_{LM}\}$ . For  $0 \leq i < L$ , let the number of composite branches emanating from any state  $s \in S_{iM}$  be  $2^{\lambda_{iM}}$ , (it is the same for any state in  $S_{iM}$ ). The ordered sequence  $\{\lambda_0, \lambda_1, \dots, \lambda_{(L-1)M}\}$  is called diverging branch profile (DBP). Then  $\delta$  and  $\lambda$  are related as follows:

$$\delta_{iM} = s_{(i-1)M} + \lambda_{(i-1)M} - s_{iM}. \quad (1)$$

Based on the theory of  $L$ -section trellises [9], it can be shown that

$$\lambda_{(i-1)M} = \dim(C_{(i-1)M, N}) - \dim(C_{iM, N}) - P_i \quad (2)$$

which implies that  $\lambda_{(i-1)M}$  equals the numbers of rows of a trellis oriented generator matrix of  $C$  whose leading 1 occurs among the positions  $\{(i-1)M, (i-1)M + 1, \dots, iM - 1\}$  and whose span is not contained in the  $i$ th section. These dimensions can be easily determined from the trellis oriented generator matrix of the code [9], [16], [20]. The two sequences,  $\{\delta_M, \delta_{2M}, \dots, \delta_{LM}\}$  and  $\{\lambda_0, \lambda_1, \dots, \lambda_{(L-1)M}\}$  provide a measure of the state connectivity of an  $L$ -section minimal trellis. In IC implementation of a Viterbi decoder,  $\delta_{iM}$  is called a radix number.

### B. Parallel Trellises

Let  $G$  be the trellis oriented generator matrix of an  $(N, K)$  linear block code  $C$  [4]. Let  $\mathbf{r} = (r_1, r_2, \dots, r_N)$  be a typical row of  $G$ . Then, we define the *span* of  $\mathbf{r}$ , denoted  $\text{span}(\mathbf{r})$ , to be the smallest interval  $[i, j]$ ,  $1 \leq i \leq j \leq N$  which contains all the nonzero elements of  $\mathbf{r}$ . For a row  $\mathbf{r}$  whose span is  $[i, j]$  we also define an *active span* of  $\mathbf{r}$ , denoted  $\text{aspan}(\mathbf{r})$ , as  $[i, j - 1]$  if  $i < j$  and  $\text{aspan}(\mathbf{r}) = \emptyset$  if  $i = j$ . The trellis oriented matrix has the following properties: 1) The leading one of every row occurs in an earlier position than the leading one of the row below it and 2) The trailing one of every row occurs at a different position from the trailing one of every other row. Any other trellis oriented matrix for  $C$  has the same set of *row spans* although the rows themselves may be different [20]. Let  $T$  be the minimal  $N$ -section trellis for  $C$ .

Given the trellis oriented generator matrix of a code, the state space dimension at any position  $l$  is just equal to the number of rows whose active span contain  $l$  [20]. For example, consider the following trellis oriented generator matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \mathbf{r}_1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \mathbf{r}_2 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \mathbf{r}_3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \mathbf{r}_4 \end{pmatrix}$$

for which  $\text{aspan}(\mathbf{r}_1) = [1, 3]$ ,  $\text{aspan}(\mathbf{r}_2) = [2, 6]$ ,  $\text{aspan}(\mathbf{r}_3) = [3, 5]$  and  $\text{aspan}(\mathbf{r}_4) = [5, 7]$ . For each  $l$ ,  $0 \leq l \leq 8$ , counting the number of rows which are active at that  $l$  yields the state complexity profile (SCP),  $\{0, 1, 2, 3, 2, 3, 2, 1, 0\}$ . For  $0 \leq l \leq N$ , let  $s_l(C)$  denote the dimension of the  $l$ th state space of  $C$ . Let  $s_{\max}(C)$  be the maximum among the state space dimensions. Define the nonempty set

$$I_{\max}(C) = \{l: s_l(C) = s_{\max}(C)\}. \quad (3)$$

Suppose we choose a subcode  $C'$  of  $C$  such that  $\dim(C') = \dim(C) - 1$  and the set of coset representatives  $[C/C']$  is generated by the single row  $\mathbf{r} \in G$ . From the above statement about  $s_l(C)$ , it is clear that  $s_l(C') = s_l(C) - 1$  for exactly those  $l$  where  $\mathbf{r}$  is active. i.e.,  $l \in \text{aspan}(\mathbf{r})$ . For other positions  $l \notin \text{aspan}(\mathbf{r})$  we have  $s_l(C') = s_l(C)$ . Hence, we have the following proposition.

**Lemma 1:** If there exists a row  $\mathbf{r}$  in the trellis oriented generator matrix  $G$  for the code  $C$  such that  $\text{aspan}(\mathbf{r}) \supseteq I_{\max}(C)$ , then we can form a subcode  $C'$  of  $C$  generated by  $G - \{\mathbf{r}\}$  such that  $s_{\max}(C') = s_{\max}(C) - 1$  and  $I_{\max}(C') \supseteq I_{\max}(C)$ . ■

In fact  $I_{\max}(C') = I_{\max}(C) \cup \{l: s_l(C) = s_{\max}(C) - 1, l \notin \text{aspan}(\mathbf{r})\}$ . Since  $G$  is a trellis-oriented generator matrix,  $G' = G - \{\mathbf{r}\}$  is also trellis-oriented. We can apply the above proposition again to  $C'$  if there exists a row  $\mathbf{r}' \in G'$  with  $\text{aspan}(\mathbf{r}') \supseteq I_{\max}(C')$ . This yields a subcode  $\tilde{C}$  with dimension smaller by one and  $s_{\max}(\tilde{C}) = s_{\max}(C') - 1$ . If no such row  $\mathbf{r}'$  exists, the proposition cannot be applied and the recursion stops. The above proposition can be generalized.

Let  $R(C)$  be the following subset of rows of  $G$ :

$$R(C) = \{\mathbf{r} \in G: \text{aspan}(\mathbf{r}) \supseteq I_{\max}(C)\}. \quad (4)$$

Let  $\rho = |R(C)|$  where  $|Q|$  denotes the cardinality of any finite set  $Q$ .

**Theorem 1:** With  $R(C)$  defined as above and  $\rho = |R(C)|$ , let  $1 \leq \rho' \leq \rho$ . There exists a subcode  $C'$  of  $C$  such that  $s_{\max}(C') = s_{\max}(C) - \rho'$  and  $\dim(C') = \dim(C) - \rho'$  if and only if there exists a subset  $R' \subseteq R(C)$  consisting of  $\rho'$  rows of  $R(C)$  such that for every  $l$  satisfying  $s_l(C) > s_{\max}(C')$ , there exist at least  $s_l(C) - s_{\max}(C')$  rows in  $R'$  whose active spans contain  $l$ . The set of coset representatives  $[C/C']$  is generated by  $R'$ .

**Proof:** Suppose  $R' = \{\mathbf{r}'_1, \dots, \mathbf{r}'_{\rho'}\}$  satisfies the conditions in the hypothesis. Since  $R' \subseteq R(C)$ ,  $I_{\max}(C) \subseteq \text{aspan}(\mathbf{r}'_i)$  for  $1 \leq i \leq \rho'$ . Consider the subcode generated by  $G - R'$ . For those  $l \in I_{\max}(C)$ , we can determine  $s_l(C')$  by counting the number of rows  $\mathbf{r} \in (G - R')$  that are active at the position  $l$ . But this number is exactly less than  $s_{\max}(C)$



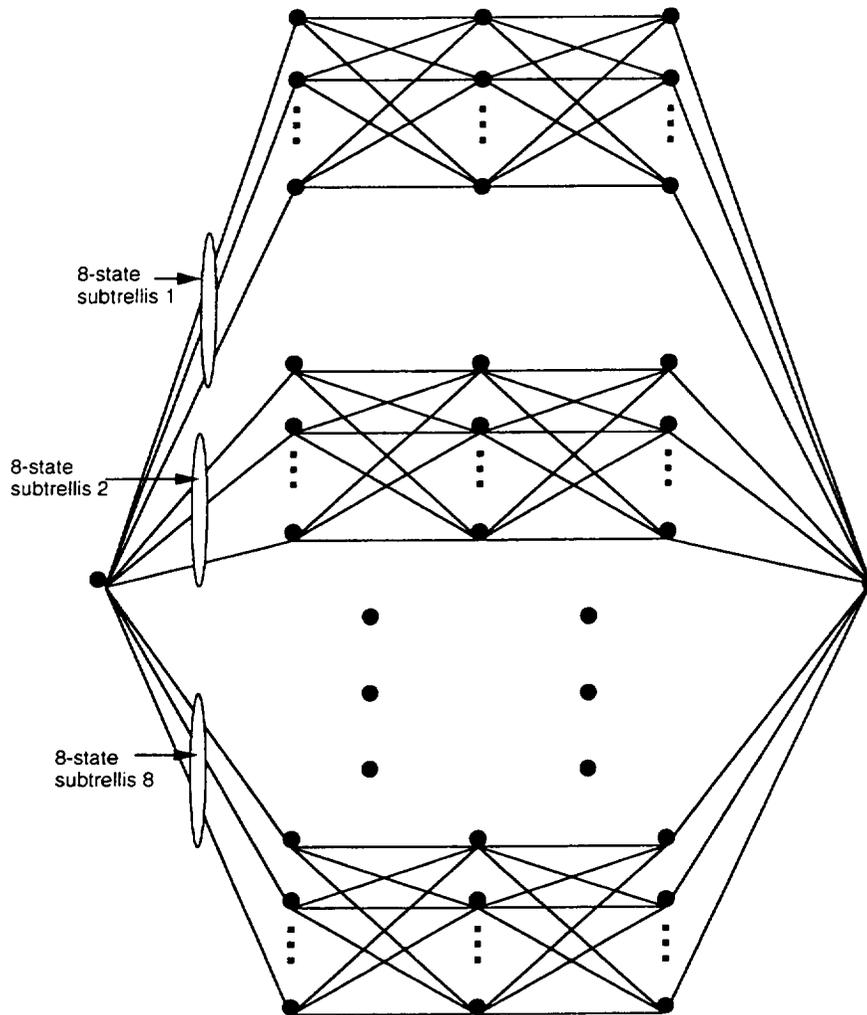


Fig. 2. A four-section, 64-state trellis for the (32,16,8) RM code composed of eight parallel and isomorphic eight-state subtrellises.

the subset of two rows having span [8, 25] and [10], [23] satisfy the conditions in Theorem 1.

This decomposition of a trellis into parallel and structurally identical subtrellises of smaller state complexity without cross-connections between them has significant advantages for IC implementation of Viterbi decoding. Identical Viterbi decoders of much simpler complexity can be devised to process the subtrellises independently in parallel without internal communications (or information transfer) between them. Internal information transfer limits the decoding speed [28], [30]. Furthermore, the number of computations to be carried out per subtrellis is much smaller than that of a fully connected trellis. As a result, the parallel structure not only simplifies the decoding complexity but also speeds up the decoding process. For example, the (32,16,8) extended and permuted BCH code (also a RM code) has a four-section trellis diagram of 64 states. It can be decomposed into eight parallel and structurally identical eight-state subtrellises without cross-connections between them as shown in Fig. 2. As a result, eight identical eight-state Viterbi decoders can be devised to process the decoding in parallel. An IC implementation of a Viterbi decoder for this code using a 0.8  $\mu\text{m}$  CMOS technology

has been recently completed at the University of Hawaii VLSI Design Center. The decoder is implemented in Xilinx field programmable gate array (FPGA) chips [29]. The decoder is capable of operating at a speed of 200 Mbps. Custom design of this decoder using 0.5 micron CMOS technology can achieve a decoding speed of 600 Mbps or higher.

### III. TRELLISES OF BCH AND RM CODES OF LENGTHS 32, 64

Based on the theory developed in the previous section, an analysis of the parallel structure of the trellises for RM and extended binary BCH codes was carried out. The degree of parallelism and the state complexity both depend on the sectionalization of the trellis. In general, it is known that as the number of sections decreases, the state complexity also decreases but the branch complexity in each section increases. We consider all possible uniform sectionalizations in which the number of parallel branches between two connected states is at most two. For example, we consider only 64-, 32-, 16- and eight-section trellises for the (64,42,8) RM code because the four-section trellis has 32 parallel branches between any

two adjacent connected states. The reason is that from an implementation and computational viewpoint, greater than two parallel branches between adjacent connected states are disadvantageous.<sup>1</sup> The results of this analysis are presented in Table IV. Some surprising results are observed. A 32-section trellis for the (32,11,12) extended BCH code can be constructed which consists of 512 parallel 2-state subtrellises. An eight-section trellis for the (64,42,8) RM code can be constructed which consists of 128 parallel 64-state subtrellises. These results do not follow from the squaring construction for RM codes or other previously published approaches. They also provide the designer a wide range of choices for trellises from which to choose. In the tables for each code and for each possible choice of the number of sections  $L$ , the logarithm to base two of the maximum number of parallel subtrellises that can be obtained without exceeding the number of states in the minimal trellis is denoted  $P_{\max,L}$ . The maximum state space dimension of the  $L$ -section subtrellis for the subcode  $C'$  is denoted  $s_{\max,L}(C')$ . The best known order of bit positions with respect to state complexity of BCH codes of length 64 presented in [12], [25] was used to produce the tables.

#### IV. ISSUES IN THE IC IMPLEMENTATION OF AN $L$ -SECTION TRELLIS-BASED VITERBI DECODER

In this section, five key factors affecting the decoding speed of a Viterbi decoder based on the minimal and nonminimal trellis are examined. The nonminimal trellis structure presented in this paper reduces the internal communication and allows independent parallel processing of the subtrellises while decreasing the complexity of a Viterbi decoder IC. We substantiate this claim through analysis in the following subsections.

##### A. Effective Computational Complexity of $L$ -Section Trellis

We consider a Viterbi decoder IC based on an  $L$ -section trellis with  $M$  bits/section for a  $(LM, K, d_{\min})$  block code  $C$ . While many VLSI structures have been described for a Viterbi decoder [26], [27], [32], the most widely implemented structure is based on ACS where each abstract state in the trellis diagram manifests itself as a physical ACS circuit on the IC and the same ACS's are repeatedly used for all depths in the trellis. The ACS's can be labeled ACS- $i$  for  $0 \leq i < 2^{s_{\max,L}(C)}$ .

Let  $\gamma_i$  be the time required to process section- $i$  of the trellis. At time  $t = 0$ , the metrics of the ACS circuits corresponding to the originating state of each parallel subtrellis are initialized to zero. After  $\gamma_1$  units of time, at  $t = \gamma_1$ , the ACS- $i$  corresponding to state  $s_i$  at the end of section-1 for  $0 \leq i < |S_M(C)|$ , has the metric of state  $s_i \in S_M$ . The index of the surviving branch into  $s_i$  is also stored in ACS- $i$ . Continuing in this way, at time  $t = \gamma_1 + \dots + \gamma_l$ ,  $1 \leq l \leq L$ , ACS- $i$  corresponding to state  $s_i$ ,  $0 \leq i < |S_{iM}(C)|$ , will have the metric of  $s_i \in S_{iM}(C)$  and a sequence of  $l$  survivor branch indices corresponding to the most likely path from

<sup>1</sup> When there are exactly two parallel branches with complementary labels, the correlation metric for one branch is the negative of other and hence can be obtained by a mere sign inversion.

the originating state (of the subtrellis to which  $s_i$  belongs) to  $s_i \in S_{iM}$ .

There are as many ACS's as the maximum number of states at any depth in the  $L$ -section trellis for the linear block code. In the minimal trellis, whenever the decoder is processing the trellis at a depth at which the state size is less than the maximum state size, a number of ACS circuits are *idle* and the hardware utilization efficiency is poor. In the nonminimal  $L$ -section trellis, the utilization of the ACS circuit that exist in the IC is improved. Since all the subtrellis decoders operate independently in parallel, from the standpoint of speed, the *effective computational complexity* of decoding a single block (a received vector) is defined as the computational complexity of a single parallel subtrellis (viz. the minimal trellis for the subcode  $C'$ ) plus the cost of the final comparison among the choices (survivors) presented by each of the subtrellises. The time required for the final comparison is small relative to the time required for decoding a subtrellis and this comparison can be pipelined. Since subtrellises are processed in parallel, the speed of operation is limited only by the time required to process a subtrellis.

Note that both the minimal and nonminimal trellises require the same number of ACS circuits. However, the nonminimal trellis has a larger number of parallel subtrellises as compared to the minimal trellis (which often has none). Hence decoding using the nonminimal trellis with proper structure is faster compared to that using the minimal trellis. Therefore, a system bit rate specification which earlier could be met only by the use of some  $P$  number of Viterbi decoders operating simultaneously in parallel can be met with much fewer than  $P$  Viterbi decoders. In this manner, the effective computational complexity is a factor affecting the reduction in hardware complexity of an overall decoder.

##### B. Complexity of the ACS Circuit

The CBP defined as the number of branches merging into a state at each particular depth also affects decoding speed and implementation complexity. This is called *radix* in IC literature. Let  $\delta_{iM}(C)$ ,  $1 \leq i \leq L$ , be the CBP of the minimal trellis for  $C$  with trellis oriented generator matrix  $G$ . At depth  $l$ ,  $1 \leq l \leq L$ , the ACS circuits have to perform at least  $\delta_{iM}$  stages of a tree type [33] two-way comparisons to find the best incoming branch. Hence reduction of the converging branch profile will improve the speed of decoding and reduce the complexity of each ACS circuit. We now show that none of the components in the CBP of the nonminimal trellis is increased. As will be shown by examples in Section V, most of the components of the CBP are decreased considerably.

Consider a nonminimal trellis for  $C$  obtained as the union of two parallel subtrellises each isomorphic to the minimal trellis for  $C'$ , a subcode of  $C$  generated by  $G - \{\mathbf{r}\}$ ,  $\mathbf{r} \in G$ . Let  $\delta_{iM}(C')$ ,  $1 \leq i \leq L$ , be the CBP of the minimal trellis for  $C'$ . Recall that  $s_l(C') = s_l(C)$  if  $l \notin \text{aspan}(\mathbf{r})$  and  $s_l(C') = s_l(C) - 1$ , if  $l \in \text{aspan}(\mathbf{r})$ . By (2),  $\lambda_{(i-1)M}(C') \in \{\lambda_{(i-1)M}(C), \lambda_{(i-1)M}(C) - 1\}$ . By (1),  $\delta_{iM}(C') > \delta_{iM}(C)$  only if  $s_{(i-1)M}(C') = s_{(i-1)M}(C)$  and  $s_{iM}(C') = s_{iM}(C) - 1$ . But in this case,  $(i-1)M \notin \text{aspan}(\mathbf{r})$  and  $iM \in \text{aspan}(\mathbf{r})$ .

So  $\lambda_{(i-1)M}(\mathcal{C}') = \lambda_{(i-1)M}(\mathcal{C}) - 1$ . Therefore,  $\delta_{iM}(\mathcal{C}') = \delta_{iM}(\mathcal{C})$ .

### C. Traceback Complexity

Consider the problem of traceback to determine the best path through the trellis. In the minimal trellis, the ACS- $i$  corresponding to state  $s_i \in S_{IM}$  has to store  $\delta_{iM}(\mathcal{C}) + P_i(\mathcal{C})$  bits in order to identify which of the  $2^{\delta_{iM}(\mathcal{C})}$  composite branches merging into  $s_i$  and which of the  $2^{P_i(\mathcal{C})}$  parallel branches that form a composite branch survives. Therefore, in the minimal trellis, each ACS- $i$  needs to store  $\sum_{i=1}^L (\delta_{iM}(\mathcal{C}) + P_i(\mathcal{C})) = \dim(\mathcal{C})$  bits in order to identify sequence of surviving incoming branches. In the nonminimal trellis, the storage in number of bits required for each ACS- $i$  is  $\sum_{i=1}^L (\delta_{iM}(\mathcal{C}') + P_i(\mathcal{C}')) = \dim(\mathcal{C}')$  where  $\mathcal{C}'$  is the subcode of  $\mathcal{C}$  corresponding to the subtrellis. Since  $\dim(\mathcal{C}') = \dim(\mathcal{C}) - P_{\max,L}(\mathcal{C})$ , the ACS's in the nonminimal trellis design require less storage than in the minimal trellis. The combined savings in storage in all the  $2^{s_{\max,L}(\mathcal{C})}$  ACS circuits is significant.

### D. ACS - Connectivity

The basic operations performed by an ACS circuit are: addition of branch metrics of the incoming branches to the state metrics of the corresponding originating states, comparison of the resulting sums to find the best, selection of the surviving sum as the new state metric and the corresponding surviving branch label. The ACS-array architecture is dominated by the area required by the interconnections to transfer the state metrics [27]. For a state  $s_i \in S_{IM}$ ,  $0 \leq i < 2^{s_i(\mathcal{C})}$ ,  $0 \leq l < L$ , let  $A_l(s_i)$  denote the set of states in  $S_{(l+1)M}$  that are adjacent to  $s_i$ . Let  $A_l(s_i) = \phi$  if  $i \geq |S_{IM}|$ . Then in the ACS-array implementation of the Viterbi decoder based on the minimal trellis, a path to transfer the state metric must exist between ACS- $i$  and all ACS circuits that correspond to states in

$$A_0(s_i) \cup A_1(s_i) \cup \dots \cup A_{(L-1)}(s_i).$$

The above set defines the *connectivity* of ACS- $i$  in the ACS array corresponding to state  $s_i \in S_{IM}$ . The connectivity of the ACS's corresponding to states in the minimal trellis results in a large amount of area in the VLSI chip being used for wiring [26], [27]. On the contrary, in the implementation of a Viterbi decoder based on the nonminimal trellis, the ACS circuits can be divided into blocks [31] such that the ACS's corresponding to states in a single subtrellis form a block. A particular ACS- $i$  needs to transfer its metric only to a subset of ACS's within its own block. This reduced connectivity results in a reduction of hardware complexity and wiring area. The maximum connectivity of ACS- $i$  is upper-bounded by  $s_{\max,L}(\mathcal{C}')$  in the nonminimal trellis implementation.

### E. Branch Complexity

The number of distinct branch metrics that have to be computed in section- $i$  of the trellis is a property of the code and is *unaltered* by the parallelization of the trellis. Most IC decoders have a branch metric computational unit where all the

branch metrics are calculated and then transferred to the ACS circuits [26], [27]. Because of the interconnection of branches between states in the trellis, routing the branch metrics to each of the ACS circuits requires a large amount of chip area. The trellises we describe show improvement over the minimal  $L$ -section trellis on this count because each subtrellis requires only a subset of the set of branch metrics in section- $i$  of the trellis.

Parallelization of the minimal trellis as described in Section II may lead to a larger number of total computations being performed in decoding. The number of emanating branches in section- $(l+1)$  is  $|S_{IM}|2^{\lambda_{lM}}$  which may be larger than the corresponding product for the minimal trellis for some values of  $l$ ,  $0 \leq l < L$ . However, as explained above, the hardware complexity of the decoder is not affected. We illustrate the reason with an example: The RM (64,42,8) code has a minimal trellis with the  $s_{IM} + \lambda_{IM}$  sequence of  $\{7, 13, 16, 16, 16, 16, 13, 7\}$ . The same sequence for the nonminimal trellis is  $\{13, 16, 16, 16, 16, 16, 13\}$  which is larger at positions  $\{0, 1, 6, 7\}$ . Consider the case when  $l = 1$  (other cases are similar). In section 2 of the minimal trellis, each of the 128 ACS's corresponding to states at the end of section 1 has 64 branches emanating from it. In section 2 of the nonminimal trellis, each of the 8192 ACS's has eight branches emanating from it. Hence, the number of operations performed per ACS are fewer in the nonminimal trellis. Hence larger values of  $|S_{IM}|2^{\lambda_{lM}}$  represent larger number of operations performed simultaneously in parallel by all the ACS's in the nonminimal trellis.

## V. EXAMPLES

Consider the (32,21,6) extended and permuted BCH code. The minimal four-section, 8-bits/section trellis has SCP  $\{0, 7, 9, 7, 0\}$ . A nonminimal trellis four-section trellis can be obtained as the union of 32 parallel isomorphic subtrellises each having SCP  $\{0, 4, 4, 4, 0\}$ . Thus, Viterbi decoder implementations using the ACS-array architecture for both trellises will require 512 ACS circuits. However, in the minimal trellis, each ACS will require the capability of choosing the best among 64 incoming branches whereas the corresponding number is only 16 in the nonminimal trellis. The problem of routing metrics is also much reduced since the connectivity of ACS-0 is 128 and that of ACS- $i$  is at least 64 for  $1 \leq i \leq 511$  while the maximum connectivity of any ACS in the nonminimal trellis is only 16. The structural parameters of each of these trellises are summarized in Tables II and III.

Assuming each real number to be quantized to 8-bits the VLSI layouts of a radix-8 ACS and a radix-16 ACS were generated. A modified form of the bit-level pipelined ACS architecture [33] was used for the ACS's. The area required for the radix-16 ACS was 2.7 times that required for the radix-8 ACS. Assuming a factor of 2.5 increase in area per doubling of the radix, we see that 128 ACS's in the minimal trellis have an area 6.25 times larger than their counterparts in the nonminimal trellis implementation. The remaining ACS's require the same area. We see that the device area is reduced by adopting the proposed trellis architecture. Furthermore, the

TABLE II  
PARAMETERS OF FOUR-SECTION TRELLIS OF  
(32,21,6) EXTENDED AND PERMUTED BCH CODE

$i$	0	1	2	3	4
SCP	0	7	9	7	0
CBP	-	0	4	6	7
EBP	7	6	4	0	-

$i = \text{ACS-}\#$	Connectivity of ACS- $i$
0	128
1 - 511	64

TABLE III  
PARAMETERS OF FOUR-SECTION TRELLIS OF (32,16) SUBCODE  
OF THE (32,21,6) EXTENDED AND PERMUTED BCH CODE

$i$	0	1	2	3	4
SCP	0	4	4	4	0
CBP	-	0	4	4	4
EBP	4	4	4	0	-

$i = \text{ACS-}\#$	Connectivity of ACS- $i$
0 - 511	16

reduction in ACS-connectivity will yield significant reduction in wiring area. The savings in hardware complexity and increase in speed due to the nonminimal trellis approach easily overcomes the extra cost of the final comparison among the 32 choices (one from each of the subtrellises) to find the best codeword.

## VI. TRELLIS FOR A (64,40,8) SUBCODE OF RM (64,42,8)

A (64,40,8) subcode of the RM (64,42,8) code is proposed to NASA for usage as inner code in a concatenated coding system with the NASA standard (255,223,33) Reed-Solomon code as outer code [31]. This RM subcode achieves a 5.3 dB coding gain over uncoded binary phase shift keying (BPSK) at the bit-error rate (BER) of  $10^{-6}$ . The required speed of decoding is  $960 \times 10^6$  BPSK symbols/s which translates to an information bit rate of 600 Mbps. The coding gain is 0.5 dB less than the coding gain of a similar scheme with the same outer code but the NASA standard rate-1/2, 64-state convolutional code [34] as the inner code. However the (64,40) RM subcode has a higher rate of 0.626 b/symbol than that of the convolutional code and thus requires lesser bandwidth. More significant is the fact that a Viterbi decoder for the (64,40,8) inner code can be designed to operate at higher data rates than that for the convolutional code using the parallelism of the trellis of the RM subcode. The trellis for the NASA standard 64-state convolutional code does not consist of parallel subtrellises.

Let  $C$  denote the RM (64,42,8) code and  $\tilde{C}$  a (64,40) subcode of  $C$ . If the  $L$ -section trellis on which decoding is based is composed of a union of  $P$  parallel isomorphic subtrellises then, the effective computational complexity denoted  $A_{\text{eff}}(L)$  is merely that of a single subtrellis plus the cost of obtaining the final decision by comparing outputs of each of the  $P$  Viterbi decoders. The value of  $L$  which minimizes  $A_{\text{eff}}(L)$  with the constraint that the  $L$ -section trellis  $\tilde{T}$  have a maximum

TABLE IV  
MAXIMUM PARALLELIZATION OF TRELLISES FOR  
ALL RM AND BCH CODES OF LENGTH 32,64

		No of Sections $L$						
		64	32	16	8	4	2	
1	RM(32,6,16)	$P_{\text{max},L}(T)$		4	4	4	3	4
		$s_{\text{max},L}(C')$		1	1	1	1	0
2	BCH(32,11,12)	$P_{\text{max},L}(T)$		9	9	9	7	
		$s_{\text{max},L}(C')$		1	1	1	2	
3	RM(32,16,8)	$P_{\text{max},L}(T)$		5	4	5	3	
		$s_{\text{max},L}(C')$		4	4	3	3	
4	BCH(32,21,6)	$P_{\text{max},L}(T)$		2	4	4	5	
		$s_{\text{max},L}(C')$		8	6	6	4	
5	RM(32,26,4)	$P_{\text{max},L}(T)$		1	1	2		
		$s_{\text{max},L}(C')$		4	4	3		
6	RM(64,7,32)	$P_{\text{max},L}(T)$	5	5	5	5	4	5
		$s_{\text{max},L}(C')$	1	1	1	1	1	0
7	RM(64,10,28)	$P_{\text{max},L}(T)$	10	10	10	10	10	10
		$s_{\text{max},L}(C')$	0	0	0	0	0	0
8	BCH(64,16,24)	$P_{\text{max},L}(T)$	14	14	14	12	13	14
		$s_{\text{max},L}(C')$	1	1	1	2	1	0
9	BCH(64,18,22)	$P_{\text{max},L}(T)$	16	16	16	14	16	16
		$s_{\text{max},L}(C')$	1	1	1	2	2	2
10	RM(64,22,16)	$P_{\text{max},L}(T)$	9	9	8	9	6	
		$s_{\text{max},L}(C')$	5	5	5	4	4	
11	BCH(64,24,16)	$P_{\text{max},L}(T)$	11	11	10	11	8	
		$s_{\text{max},L}(C')$	5	5	5	4	4	
12	BCH(64,30,14)	$P_{\text{max},L}(T)$	15	13	14	11	14	
		$s_{\text{max},L}(C')$	6	7	6	7	4	
13	BCH(64,36,12)	$P_{\text{max},L}(T)$	10	9	10	9	8	
		$s_{\text{max},L}(C')$	10	10	9	8	8	
14	BCH(64,39,10)	$P_{\text{max},L}(T)$	7	8	9	10	11	
		$s_{\text{max},L}(C')$	13	12	11	9	8	
15	RM(64,42,8)	$P_{\text{max},L}(T)$	5	6	5	7		
		$s_{\text{max},L}(C')$	9	8	8	6		
16	BCH(64,45,8)	$P_{\text{max},L}(T)$	2	3	4	4		
		$s_{\text{max},L}(C')$	12	11	10	9		
17	BCH(64,51,6)	$P_{\text{max},L}(T)$	1	1	1	2		
		$s_{\text{max},L}(C')$	11	11	11	10		
18	RM(64,57,4)	$P_{\text{max},L}(T)$	1	1	1	2		
		$s_{\text{max},L}(C')$	5	5	5	4		

state complexity not greater than  $s_{\text{max},L}(\tilde{C})$  [which is different from  $s(C)$ ] is determined. Note that  $s_{\text{max},L}(\tilde{C})$  is a function of the choice of the subcode and we will choose that subcode which has the least  $s_{\text{max},L}(\tilde{C})$  for each  $L$ . The complexity of each addition, subtraction and comparison is assumed to be equal to one addition equivalent operation.

In the following, the trellis diagrams of various sectionizations for this RM subcode are given. Their effective computational complexities are computed.

### A. $L = 4, M = 16$

Let  $C_0 = (16, 15, 2)$ ,  $C_1 = (16, 11, 4)$ , and  $C_2 = (16, 5, 8)$  be the corresponding RM codes,  $G_i$  a generator matrix of  $C_i$  and  $G_{i/j}$  a generator matrix for the set of coset representatives  $[C_i/C_j]$ . Let  $\times$  denote the Kronecker product. For  $L = 4$ , the RM (64,42) code has a minimal trellis corresponding to the 2-level squaring construction with a state complexity profile (SCP)  $\{0, 10, 10, 10, 0\}$  ( $s_{\text{max},4}(C) = 10$ ) and trellis-oriented

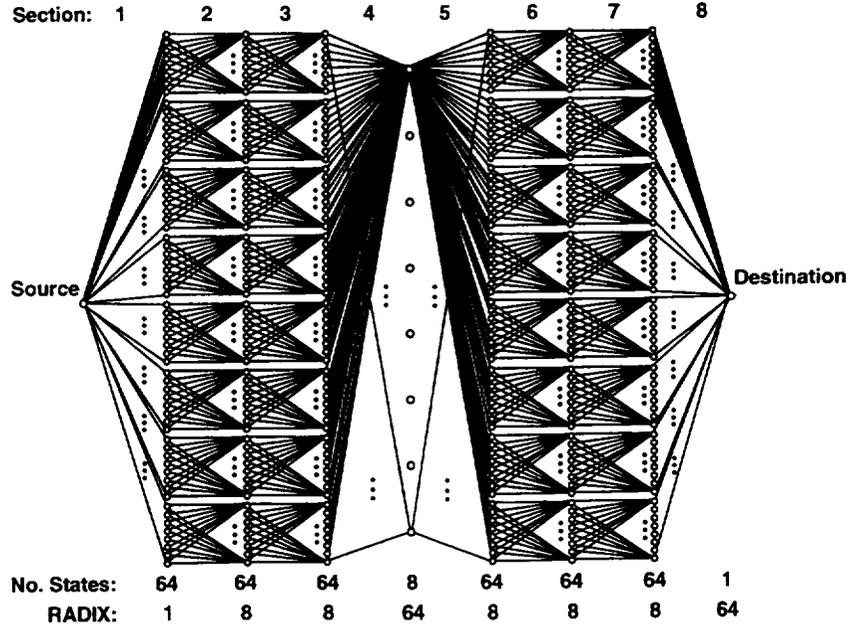


Fig. 3. An eight-section, 64-state subtrellis for the (64,35,8) subcode of the (64,40,8) RM subcode.

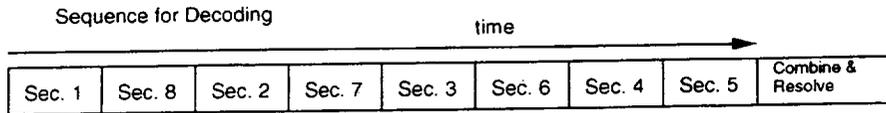


Fig. 4. Sequence for decoding using concurrent bidirectional execution sequence.

generator matrix

$$\begin{aligned}
 \mathbf{G} &= (1 \ 1 \ 1 \ 1) \otimes \mathbf{G}_{0/1} \\
 &+ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \otimes \mathbf{G}_{1/2} \\
 &+ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \mathbf{G}_2. \quad (7)
 \end{aligned}$$

In order to obtain a (64,40) subcode  $\tilde{\mathcal{C}}$ , one can delete any two of the 64 rows above giving a generator matrix for  $\tilde{\mathcal{C}}$ . The maximum state space complexity  $s_{\max,4}(\tilde{\mathcal{C}})$  of the resulting code depends on which two rows we delete. It is easy to see that in order to have the least  $s_{\max,4}(\tilde{\mathcal{C}})$  which equals 8 we must delete any two of the four rows among  $(1111) \otimes \mathbf{G}_{0/1}$  obtaining an SCP of  $\{0, 8, 8, 8, 0\}$  ( $s_{\max,4}(\tilde{\mathcal{C}}) = 4$ ). Using the theory developed earlier, it can be seen that we can obtain at most four parallel subtrellises in any four-section trellis for  $\tilde{\mathcal{C}}$  without exceeding the allowable  $s_{\max,4}$  of eight. The effective computational complexity may be computed to give  $A_{\text{eff}}(4) = 39\,682$  addition equivalent operations for the four-section trellis.

B.  $L = 8, M = 8$

Let  $\mathcal{C}_0 = (8, 8, 1), \mathcal{C}_1 = (8, 7, 2), \mathcal{C}_2 = (8, 4, 4), \mathcal{C}_3 = (8, 1, 8)$  be RM codes. For  $L = 8$ , the RM (64,42,8) code

has a minimal trellis (with two parallel subtrellises) corresponding to the 3-level squaring construction with a SCP  $\{0, 7, 10, 13, 10, 13, 10, 7, 0\}$  ( $s_{\max,8}(\mathcal{C}) = 13$ ) with trellis oriented generator matrix

$$\begin{aligned}
 \tilde{\mathbf{G}} &= (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ \mathbf{r}_0^0) \otimes \mathbf{G}_{0/1} \\
 &+ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \mathbf{r}_0^1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \mathbf{r}_1^1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \mathbf{r}_2^1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \mathbf{r}_3^1 \end{pmatrix} \otimes \mathbf{G}_{1/2} \\
 &+ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \otimes \mathbf{G}_{2/3} + \mathbf{I}_8 \otimes \mathbf{G}_3. \quad (8)
 \end{aligned}$$

The (64,40) subcode  $\tilde{\mathcal{C}}$  with the best SCP is obtained by deleting the rows  $\mathbf{r}_0^0 \otimes \mathbf{G}_{0/1}$  and any one among the three rows  $\mathbf{r}_1^1 \otimes \mathbf{G}_{1/2}$ . This code  $\tilde{\mathcal{C}}$  has SCP  $\{0, 6, 8, 11, 8, 11, 8, 6, 0\}$  ( $s_{\max,8}(\tilde{\mathcal{C}}) = 11$ ). Repeating a similar analysis, it is seen that one can obtain at most 32 parallel subtrellises in any eight-section trellis for  $\tilde{\mathcal{C}}$  without exceeding the maximum allowable state space complexity of  $s_{\max,8}(\tilde{\mathcal{C}}) = 11$ . Each subtrellis has the SCP  $\{0, 6, 6, 6, 3, 6, 6, 6, 0\}$  and from knowledge of its trellis structure the effective complexity is  $A_{\text{eff}}(8) = 12\,822$  addition equivalent operations.

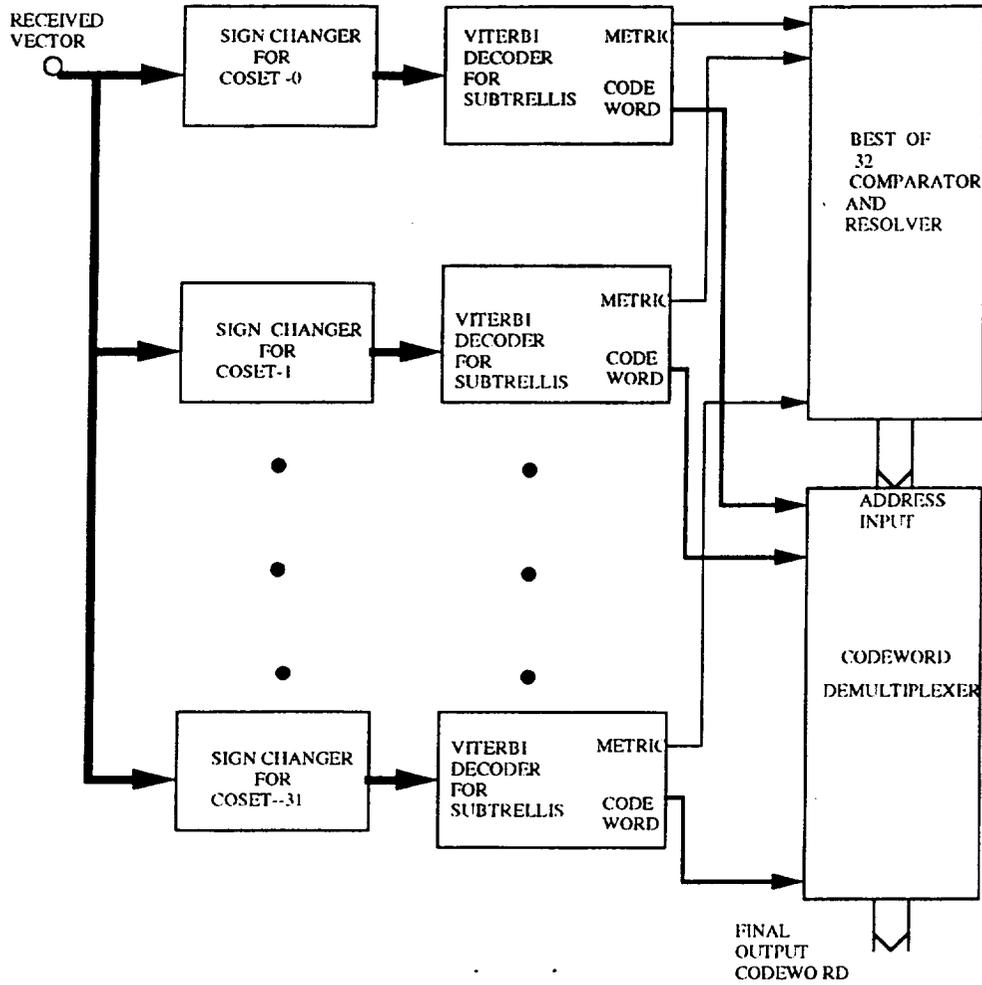


Fig. 5. Block diagram of overall decoder with 32 Viterbi decoders.

C.  $L = 16, M = 4$

Let  $C_0 = C_1 = (4, 4, 1), C_2 = (4, 3, 2), C_3 = (4, 1, 4), C_4 = (4, 0, \infty)$  be RM codes. For  $L = 16$ , the RM (64,42,8) code has a trellis oriented generator matrix given by

$$G = G_{\text{RM}(16,5,8)} \otimes G_{1/2} + G_{\text{RM}(16,11,4)} \otimes G_{2/3} + G_{\text{RM}(16,15,2)} \otimes G_{3/4} \quad (9)$$

where  $G_{\text{RM}(n,k,d)}$  denotes a trellis oriented generator matrix for the  $(n, k, d)$  RM code. For  $L = 16$ , the RM (64,42,8) code has a minimal trellis (with no parallel subtrellises) corresponding to the four-level squaring construction with a SCP  $\{0, 4, 7, 10, 10, 13, 13, 13, 10, 13, 13, 13, 10, 10, 7, 4, 0\}$  ( $s_{\text{max},16}(\tilde{C}) = 13$ ). The (64,40) subcode  $\tilde{C}$  with the best SCP is generated by  $\tilde{G} = G - \{\mathbf{r}_1^1 \otimes G_{1/2}, \mathbf{r}_2^1 \otimes G_{1/2}\}$  where  $\mathbf{r}_1^1$  and  $\mathbf{r}_2^1$  are the two rows with span [2], [15] and [3], [14] in the trellis oriented generator matrix for RM (16,11,4). The SCP of the minimal trellis for  $\tilde{C}$  is  $\{0, 4, 6, 8, 8, 11, 11, 11, 8, 11, 11, 11, 8, 8, 6, 4, 0\}$  ( $s_{\text{max},16}(\tilde{C}) = 11$ ). By analysis, one can obtain at most 8 parallel subtrellises in any 16-section trellis for  $\tilde{C}$  without exceeding the allowable  $s_{\text{max},16}(\tilde{C})$  of 11. Each subtrellis has SCP  $\{0, 4, 6, 8, 8, 8, 8, 8, 5, 8, 8, 8, 6, 8, 6, 4, 0\}$ . The resulting

effective computational complexity is  $A_{\text{eff}}(16) = 23174$  addition equivalent operations.

D.  $L = 32, M = 2$  and  $L = 64, M = 1$

When  $L = 32, s_{\text{max},32}(\tilde{C}) = 12$ . The maximum number of parallel isomorphic subtrellises possible without exceeding the allowable  $s_{\text{max},32}(\tilde{C}) = 12$  in any 32-section trellis for the (64,40) subcode  $\tilde{C}$  is at most 4. So  $A_{\text{eff}}(32) \geq 37476$ . When  $L = 64, s_{\text{max},64}(\tilde{C}) = 12$ . Furthermore, no parallel subtrellises are possible without exceeding the allowable  $s_{\text{max},64}(\tilde{C}) = 12$ . Hence  $A_{\text{eff}}(64) = 198000$ .

From the above analysis, we see that the eight-section trellis for the (64,40) RM subcode results in the least effective complexity. A VLSI implementation of a high-speed decoder for the (64,40) RM subcode is under way. The decoder is based on the eight-section trellis which is a union of 32 parallel isomorphic subtrellises with a maximum of 64-states each. A schematic of the subtrellis is shown in Fig. 3. Note that the last four sections of the subtrellis form a mirror image of the first four sections. This structure allows us to perform bidirectional decoding from both ends of the subtrellis simultaneously [10], [31], [35]. Sections one through four and sections eight through five (in reverse order) are processed

at the same time and path information corresponding to the most likely paths into the center eight states which are the destination states are stored. The two path metrics (one from each side) at a center state are then added. This gives path metrics of eight final survivors and the path with the largest path metric is the most likely path through the subtrellis. Since the resolution is done at the center of the subtrellis, the bottleneck of decoding caused by the large radix at the center states is avoided. This bidirectional decoding can be achieved by either using two identical subtrellis decoders working from both directions or using only one decoder to process the subtrellis in a concurrent bidirectional execution sequence as shown in Fig. 4. The second approach simply exploits the use of pipelining in the ACS implementation and the mirror symmetry of the subtrellis about the center axis. The bidirectional decoding results in advantages in speed and implementation. A block diagram for the overall decoder is shown in Fig. 5. We further note that sections two, three, four, five, six, and seven of each subtrellis decompose into eight parallel, eight-state, fully connected isomorphic sub-subtrellises as depicted in Fig. 3. This fact can be used to further reduce implementation complexity and increase the decoding speed.

## VII. CONCLUSION

We have presented an approach for decomposing the minimal trellis of a binary linear block code into a nonminimal trellis composed of parallel components. This approach allows parallel processing of the subtrellises and does not increase the maximum number of states. Hence, it has significant speed advantage. In addition, it also reduces the IC area requirements. Given a linear block code, we have estimated the limits to the benefits of this approach and its dependence on the uniform sectionalization of the trellis. The branch complexity of the nonminimal trellis relative to the minimal trellis can be larger in some sections. However, this does not increase the hardware complexity. Since the application of this method depends only on the generator matrix of the code, it can be applied to arbitrary linear block codes.

## ACKNOWLEDGMENT

The authors would like to thank T. Fujiwara of Osaka University for providing them with the generator matrices of some extended BCH codes with the best known order of bit positions with respect to trellis state complexity and C. W. Chu and E. Nakamura of the University of Hawaii for helpful discussions relating to the VLSI aspects of this paper. They also thank the reviewers for their many helpful suggestions and constructive criticism.

## REFERENCES

- [1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear block codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, 1974.
- [2] J. K. Wolf, "Efficient maximum likelihood decoding on linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan. 1978.
- [3] J. L. Massey, "Foundation and methods of channel encoding," in *Proc. Intl. Conf. Inform. Theory Syst.*, NTG-Fachberichte, Berlin 1978.

- [4] G. D. Forney, Jr., "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.
- [5] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.
- [6] Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, 1993.
- [7] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams of binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, Jan. 1993.
- [8] ———, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 1057–1064, May 1993.
- [9] ———, "On structural complexity of the  $L$ -section minimal trellis diagrams for binary linear block codes," *IEICE Trans. Fundamentals of Electron., Commun., Computer Sci.*, vol. E76-A, no. 9, pp. 1411–1421, Sept. 1993.
- [10] ———, "On branch labels of parallel components of the  $L$ -section minimal trellis diagrams for binary linear block codes," *IEICE Trans. Fundamentals Electron., Commun., Computer Sci.*, vol. E77-A, no. 6, pp. 1058–1068, June 1994.
- [11] G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, no. 5, pp. 1491–1513, Sept. 1993.
- [12] A. Vardy and Y. Be'ery, "Maximum likelihood soft-decision decoding of BCH Codes," *IEEE Trans. Inform. Theory*, vol. 40, Mar. 1994.
- [13] G. D. Forney, Jr., "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1741–1751, Nov. 1994.
- [14] ———, "Dimension/length profiles and trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol. 40, no. 7, pp. 1753–1772, Nov. 1994.
- [15] ———, "Trellises old and new," in *Communications and Cryptography*, R. E. Blahut, D. J. Costello, U. Maurer, and T.M. Mittelholzer, Eds. Norwell, MA: Kluwer, 1994, pp. 115–128.
- [16] H. T. Moorthy and S. Lin, "On the labeling of minimal trellises for linear block codes," in *Proc. Int. Symp. Inform. Theory and Its Applications 1994*, Institution of Engineers, Australia, vol. 1, pp. 33–38.
- [17] A. Lafourcade and A. Vardy, "Asymptotically good codes have infinite trellis complexity," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 555–559, Mar. 1995.
- [18] O. Ytrehus, "On the trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 559–560, Mar. 1995.
- [19] Y. Berger and Y. Be'ery, "Trellis-oriented decomposition and trellis complexity of composite-length cyclic codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1185–1191, July 1995.
- [20] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1924–1937, Nov. 1995.
- [21] A. Lafourcade and A. Vardy, "Lower bounds on trellis complexity of block codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1924–1937, Nov. 1995.
- [22] ———, "Optimal sectionalization of a trellis," submitted for publication.
- [23] R. J. McEliece, "On the BCJR trellis for linear block codes," submitted for publication.
- [24] T. Fujiwara, H. Yamamoto, T. Kasami, and S. Lin, "A recursive maximum likelihood decoding procedure for a linear block code using a sectionalized trellis diagram and its optimization," in *Proc. 23rd Annual Allerton Conf. Commun., Control and Computing*, Allerton House, Monticello, IL, Oct. 4–6, 1995, submitted for publication.
- [25] T. Fujiwara, T. Kasami, R. M. Zaragoza, and S. Lin, "The state complexity of trellis diagrams for a class of generalized concatenated codes," submitted for publication.
- [26] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, Radix-4 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, Dec. 1992.
- [27] P. G. Gulak and T. Kailath, "Locally connected VLSI architectures for the Viterbi algorithm," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 526–537, Apr. 1988.
- [28] O. M. Collins, "The subtleties and intricacies of building a constraint length 15 convolutional decoder," *IEEE Trans. Commun.*, vol. 40, no. 12, pp. 1810–1819, Dec. 1992.
- [29] B. S. Vishwanath, "Soft-decision Viterbi decoding of the (32,16) Reed–Muller code and its VLSI implementation," M.S. thesis, Department of Electrical Engineering, University of Hawaii at Manoa, Aug. 1993.
- [30] G. Fettweis and H. Meyr, "Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck," *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 785–789, Aug. 1989.
- [31] S. Lin, G. T. Uehara, E. Nakamura, and W. P. Chu, "Circuit design approaches for implementation of a subtrellis IC for Reed–Muller subcode," NASA Tech. Rep. 96–001, Feb. 1996.

- [32] H. Thapar and J. Cioffi, "A block processing method for designing high-speed Viterbi detectors," in *Proc. ICC*, vol. 2., June 1989, pp. 1096-1100.
- [33] A. K. Yeung and J. M. Rabaey, "A 210 Mb/s Radix-4 Bit-level pipelined Viterbi decoder," in *ISSCC 1995 Dig. Tech. Papers*, San Francisco, CA, Feb. 1995.
- [34] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [35] M. Fossorier and S. Lin, "Coset codes viewed as terminated convolutional codes," submitted for publication.



**Hari T. Moorthy** (S'88-M'96) received the B.E. degree from Anna University, Madras, India, in 1989, the M.S. degree from The University of Rhode Island, Kingston, in 1992, and the Ph.D. degree from The University of Hawaii, Honolulu, in 1996, all in electrical engineering.

At Philips Research Laboratories, New York, he designed architecture and wrote test programs for Reed-Solomon decoders used in the U.S. HDTV and DBS standards. At The University of Hawaii, he is involved in VLSI design of architecture for a

very-high-speed Viterbi decoder for block codes. His main research interest is in efficient decoding algorithms for linear block codes. Other research interests are in error control coding, coded modulation, and implementation issues in communications engineering.



**Shu Lin** received the B.S.E.E. degree from the National Taiwan University, Taipei, R.O.C., in 1959, and the M.S. and Ph.D. degrees in electrical engineering from the Rice University, Houston, TX, in 1964 and 1965, respectively.

In 1965, he joined the Faculty of the University of Hawaii, Honolulu, as an Assistant Professor of Electrical Engineering. He was promoted to Associate Professor in 1969 and to Professor in 1973. In 1986, he joined Texas A&M University as the Irma Runyon Chair Professor of Electrical Engi-

neering. In 1987, he returned to the University of Hawaii and served as the Chairman of the Department of Electrical Engineering from 1989 to 1995. He spent 1978-1979 as a Visiting Scientist at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he worked on error control protocols for data communication systems. His current research areas include: algebraic coding theory, coded modulation, error control systems, and satellite communications. He has served as the Principal Investigator on 25 research grants. He was invited as a Distinguished Visiting Professor at the Nara Institute of Science and Technology, Nara, Japan, 1996. He was a research fellow of the Japan Telecommunication Advancement Organization in 1995. He has published numerous technical papers in IEEE TRANSACTIONS and other refereed journals. He is the author of the book, *An Introduction to Error-Correcting Codes* (Englewood Cliffs, NJ: Prentice Hall, 1970). He also co-authored (with D. J. Costello) the book, *Error Control Coding: Fundamentals and Applications* (Englewood Cliffs, NJ: Prentice Hall, 1982). He served as the Associate Editor for Algebraic Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY from 1976 to 1978.

Dr. Lin is a recipient of the Alexander von Humboldt Research Award, 1996. He was the Program Co-Chairman of the IEEE International Symposium on Information Theory held in Kobe, Japan, in June 1988. He was the President of the IEEE Information Theory Society in 1991.



**Gregory T. Uehara** was born in Honolulu, HI, on July 1, 1960. He received the B.S. degree in pre-engineering from The College of Idaho, Caldwell, and the B.S. degree in electrical engineering, both in 1983, and the M.S. and Ph.D. degrees, both in electrical engineering from the University of California, Berkeley, in 1989 and 1993, respectively.

He worked as a Design Engineer in mixed-signal CMOS integrated circuit design, first in the Telecommunications Group at Intel Corporation, Chandler, AZ, from 1983 to 1985, and then the Custom CMOS Circuit Design Group at MicroLinear Corporation, San Jose, CA, from 1985 to 1986. Since the fall of 1994, he has been an Assistant Professor with the Department of Electrical Engineering at the University of Hawaii, Manoa, Honolulu. He currently serves as a Consultant to Data Path Systems, Inc., Santa Clara, CA. His research at Berkely focused on CMOS mixed-signal integrated circuit implementation of high-speed magnetic-disk read channels. His current research interests include the design of high-speed analog and digital integrated circuits for communication systems and magnetic storage systems.